FAST REFERENCE GUIDE

CP/M®

Locate a complete explanation of any CP/M command, option, or syntax —in seconds

Terrence L. Disz
of
MicroComputer Education, Inc.

HAYDEN

Fast Reference Guide to CP/M®

Terrence L. Disz
MicroComputer Education, Inc.

HAYDEN BOOK COMPANY, INC.
Hasbrouck Heights, New Jersey

Acquisitions Editor: DOUGLAS McCORMICK
Production Editor: MAUREEN CONNELLY
Art Director: JIM BERNARD
Composition: ART, COPY, AND PRINT, INC.
Printed and bound by: COMMAND WEB OFFSET, INC.

Printed in the United States of America

1	2	3	4	5	6	7	8	9	PRINTING
84	85	86	87	88	89	90	91	92	YEAR

The *Fast Reference Guide to CP/M* is a convenient and comprehensive source for learning or recalling the use of CP/M commands and command parameters. It contains a summary of all CP/M commands and parameters, an explanation of both, and examples for each command.

Also included are such useful programmer tools as the CP/M memory map, a summary of DDT and ED commands, and a summary of all BDOS calls, parameters, and return codes.

Notation Conventions—Fast Reference

[ ] Brackets:

Items within brackets are optional. Enter the items, but do not enter the brackets.

( ) Parentheses:

Of the items in parentheses, one must be chosen. Defaults are underlined. Do not enter the parentheses.

UPPERCASE:

Items in UPPERCASE are to be entered as shown.

lowercase:

Indicates that the user is to substitute an appropriate value.

¦ "Or" bar:

Separates items where a choice can be made.

n Number:

Indicates that a number is required.

<cr> RETURN:

Indicates where the RETURN key is depressed.

^c CONTROL:

Indicates that the **control** key should be held down while simultaneously pressing the character given as c.

boldface:

Boldface in the examples represents the characters typed by the user. Other characters are typed by CP/M.

Transient

Indicates that the command being described is transient. The command must exist on disk as a .COM file in order to be executed.

Built-in

Indicates that the command being described is an integral part of CP/M and does not exist on disk.

d: Disk ref:

Indicates that the user is to supply a single letter disk reference. (See below.)

Disk and File References

disk references:

Under CP/M, individual disk drives are referenced by a single letter such as A or B. When used in a command to refer to a disk drive, the letter is followed by a colon (:). For example, drive A is referenced in a CP/M command as A:, and drive B as B:. The number of disk drives is dependent on the system, but the drives are always assigned letter addresses in sequential order. That is, the second drive on a system is B:, the third is C:, etc.

filenames:

In the command descriptions in this guide, references to file names are called arbitrary file names, or **afn**'s. Although afn's can be typed in either upper- or lowercase, they are always translated to uppercase by CP/M.

An **afn** is described as **[d:]filename[.suf]** where:

d: is an optional disk drive reference such as A: or B:

filename is a 1 to 8 character filename composed of alphabetic, numeric, and special characters (with the exception of < > . , ; : = ? * [ ])

.suf is an optional suffix. If supplied, the suffix must be

preceded by a period (.), and can be from 1 to 3 alphabetic, numeric, or special characters as described above.

Examples of valid afn's are:

FOO	FOO.BAR	A:FOO.BAR
#1	#1.SON	A:#1.SON

wildcards:

When specifying an afn, the wildcard operators * and ? can sometimes be used. The detailed command reference will tell, for each command, whether wildcard matches can be used in afn's for that command. The asterisk (*) matches file names from the point of the asterisk to the end of the filename or of the suffix. The question mark (?) matches any single character in its position. Examples of wildcard references in afn's are:

. matches all files

*.asm matches all files with the.asm suffix

FOO*.* matches all files beginning with FOO

FOO?.BAR matches FOO1.BAR but not FOO.BAR

CP/M Control Characters

CP/M control characters take effect while the user types the CP/M command line. Some are used for editing the line being typed while others are used to control program execution and displays.

^C

Cancels the program currently executing under CP/M.

^E

Does a carriage return on the console, but doesn't send it to CP/M. Useful when entering long commands.

^H

Deletes the previous character; moves the cursor back one space.

^I

Generates a tab character—assumed to be 8 columns.

^J

Generates a line feed, terminating console input.

^M

Generates a carriage return, terminating console input.

^P

Printer toggle switch. Pressing it once causes characters displayed on the console to be echoed to the LST: device. Pressing it again turns the function off.

^R

Erases and retypes the current CP/M command line. This is useful if you've been using the RUB or DEL editing characters.

^S

Console display toggle. Pressing it once pauses any console displays, pressing it again resumes the display.

^U

Ignores the current CP/M command line. The current line is forgotten but not erased, a # is displayed, and the cursor moves down to the start of the next line.

^X

Ignores the current CP/M command line by erasing the entire line and repositioning the cursor at the start of the line.

^Z

Terminates console input or acts as a separator in some CP/M programs.

BACKSPACE

Deletes the previous character, erases it, and moves the cursor back one position.

DEL

Deletes the previous character and echoes it to the console. (A throwback to the days of the teletype.)

RUB

Exactly the same as DEL.

CP/M Commands—Fast Reference

d:

Change current disk drive. **(Built-in)**

ASM

Execute the 8080 assembler; produce a listing and a hexadecimal object file. **(Transient)**

DDT

Execute the DDT interactive debugging program. **(Transient)**

DDT commands

A	Enter assembly language statements
D	Display memory
F	Fill memory
G	Begin program execution
I	Set up an input file block
L	List memory as assembly language statements

M	Move memory
R	Read a file set up with the I command
S	(Set) Examine and alter memory
T	Trace program execution
U	Untrace
X	E(X)amine and alter registers and CPU flags

DIR

Display a disk directory. **(Built-in)**

DUMP

Dump the contents of a file. **(Transient)**

ED

Execute the ED line editor program. **(Transient)**

ERA

Erase a file or files. **(Built-in)**

MOVCPM

Configure a new CP/M. **(Transient)**

PIP

Copy, combine, and/or transfer files among peripheral devices. **(Transient)**

REN

Rename a file. **(Built-in)**

SAVE

Save a 256-byte page from memory onto disk. **(Built-in)**

STAT

Display status information about files and peripheral devices. **(Transient)**

SUBMIT

Submit a batch of CP/M commands for execution. **(Transient)**

SYSGEN

Copy the CP/M operating system to another disk. **(Transient)**

TYPE

Type out a file onto the screen. **(Built-in)**

USER

Display or change the current user number. **(Built-in)**

XSUB

Allow buffered input in a submit file. **(Built-in)**

CP/M Commands—Detailed Reference

d:

Change current disk drive. **(Built-in)**
format: **d:**<cr>
where: **d:** is a valid disk reference.
action: Logs in the disk given as d: as the current disk. The CP/M prompt changes to reflect the new current disk.
example: A>**B:<cr>**
Logs in the B drive as the current drive. CP/M responds with: B>

ASM

Execute the 8080 assembler. **(Transient)**
format: **ASM filename**[.123]
where: **filename** is a filename of from 1 to 8 characters, the suffix of which is assumed to be .asm.

The characters 123 indicate positions where single character options may be specified.

Position **1** is used to indicate the disk drive of the source file.

Position **2** is used to indicate the destination disk drive of the HEX output file. A "z" in this position suppresses the HEX file output.

Position **3** is used to indicate the destination of the PRN output file. A disk reference in this position sends the PRN file to that disk, an "x" in this position sends the PRN file to the console, and a "z" suppresses the PRN file.

action: The 8080 assembler is loaded and assembles the source program. Depending on the program options, the assembly listing may be directed to sourcename.PRN and the hexadecimal object file to sourcename.HEX. Errors are listed on the console.

example:

ASM B:FOO

assemble the file foo.asm from the "B" disk and place the files foo.prn and foox on the "B" disk.

ASM B:CONTEST.BZZ

assemble the file contest.asm from the "B" disk; suppress the HEX file and the PRN file.

DDT

Execute the interactive debug program. **(Transient)**

format: **DDT (afn)**

where: **afn** is a program to be debugged (no wildcards allowed). If no afn is specified, DDT is loaded and waits for commands.

action: The DDT interactive debugger is loaded, loads the optional file if specified, issues the dash character prompt (-), and waits for user commands.

DDT commands: (Notice that there are no spaces between DDT commands and their parameters. All addresses are in hexadecimal)

An — Assemble statements in line.

where: **n** is a hexadecimal starting address.

DDT will prompt for the instructions to be entered starting at address n. An empty line (a <cr>) terminates the A command.

D[s[,e]] — Display memory in hexadecimal and ASCII.

where: **s** is an optional starting address
e is an optional ending address

If no ending address is specified, 16 lines are displayed. If no starting address is specified, the display begins from the current address, which is initialized to 100H. The current display address is always set to the address following the last one displayed. The **RUB** key can be used to cancel a long display.

Fs,e,c — Fill memory with a constant

where: **s** is the starting address
e is the ending address
c is the fill constant

Memory from s to e is filled with the constant c.

G[s][,b1][,b2]—Go (Resume program execution)

where: **s** is an optional starting address
b1 is a breakpoint address
b2 is a breakpoint address

Execution begins at the current address, or, if specified, at the starting address given. Execution is interrupted at either of the breakpoint addresses given.

Iafn — Insert a filename into the FCB at 5CH

where: **afn** is an arbitrary file name—no wildcards are allowed

Used with the **R** command to read files during a debug session.

L[s[,e]]—List disassembled machine code

where: **s** is an optional starting address
e is an optional ending address

If no ending address is given, 12 lines are displayed. If no starting address is given, the display starts from the current list address. At the end of a display, the current list address is set to the address following the last one displayed. The **RUB** key can be used to abort long displays.

Ms,e,d—Move a block of memory

where: **s** is the starting address
e is the ending address
d is the destination address

The block of memory from s to e is moved to location d.

R[b]—Read a file set up by the I command

where: **b** is an optional address bias. If none is specified, 0000 is assumed.

A file with a suffix of COM is assumed to be binary machine code. All other files are assumed to be in Intel hex format. As the file is loaded, the bias address is added to each address, allowing the file to be relocated anywhere in memory. (Be careful not to overwrite the system at addresses 000-0FFH.)

Ss—Examine and alter memory

where: **s** is the address to start examining.

DDT responds by displaying the address and the contents of that address location. The user may type new data and a <cr>, which alters the memory, or just a <cr>, which does not alter the displayed location. DDT then responds with the next address. The S command is terminated by entering a period (.) followed by a <cr>, or by entering invalid data.

T[n]—Trace program execution

where: **n** is the optional number of program steps to trace. The default is 1.

n program steps are executed, then a breakpoint is taken. The CPU state is printed out and the next address to execute is displayed.

U[n]—Untrace

where: **n** is the optional number of program steps to execute. 1 is assumed if n is omitted.

Executes n program steps as does the T command, but doesn't print out a trace.

X[r]—Examine and alter CPU state

where: **r** is as follows:

Omitted	— Display all registers and flags
C	— Carry flag
Z	— Zero flag
M	— Minus flag
E	— Even parity flag
I	— Interdigit carry
A	— Accumulator
B	— BC registers
D	— DE registers
H	— HL registers
S	— Stack pointer
P	— Program counter

DDT responds by displaying the register or register pair and waits for user input. A <cr> doesn't alter any values; otherwise, a complete replacement value for the register or register **pair** must be entered. That is, register C can't be altered without also entering the value for register B.

DIR

Display a disk directory. **(Built-in)**

format: **DIR ([ ] ¦ [a:] ¦ [afn])<cr>**

where: **[ ]** indicates no operands
a: is a disk reference
afn is an arbitrary file name with wildcard matching.

action: **DIR** with no operands displays the directory of the current disk.
DIR a: displays the directory of the named disk.
DIR afn displays the directory for the matching filenames.

example: A>**DIR**<cr> displays the directory on disk A.
A>**DIR B:**<cr> displays the directory on disk B.
A>**DIR *.ASM**<cr> displays the directory entries for all files on disk A that have the suffix "ASM".

DUMP

Dump the contents of a file in hexadecimal. **(Transient)**
format: **DUMP afn <cr>**
where: **afn** is an arbitrary file name—no wildcards are allowed.
action: The file contents are displayed on the console in hexadecimal, 16 bytes to a line. Long displays can be cancelled with the **RUB** key.
example: **A>DUMP B:EXAMPLE.HEX**<cr> displays the file EXAMPLE.HEX in hexadecimal.

ED

Edit a file with the ED line editor
format: **ED afn**<cr>
where: **afn** is an arbitrary file name—no wildcards.
action: The ED line editor is loaded and awaits user commands. (See ED command summary following.)

ED Command Summary—Fast Reference

nA<cr>	Append n lines to buffer (n = 0; use half of buffer)
B<cr>	Move pointer to beginning of file
-B<cr>	Move pointer to end of file
nC<cr>	Move pointer to forward n characters
nD<cr>	Delete n characters forward
E<cr>	End edit, close file, return to CP/M
nFs<cr>	Find nth occurrence of string "s"
H<cr>	End edit, move pointer to beginning of file
I<cr>	Insert text at pointer until ^z typed
Is<cr>	Insert string at pointer
nK<cr>	Kill n lines starting at pointer
nL<cr>	Move pointer n lines

nMx<cr>	Execute command string "x" n times
nNs<cr>	Global F-command—until end of file
O<cr>	Abort Ed, start over with original file
nP<cr>	List next n pages of 23 lines (n=0—current page)
Q<cr>	Quit without changing input file
Rfn<cr>	Read fn.LIB into buffer at current pointer
nSx^Zy<cr>	Substitute string "y" for next n forward occurrences of string "x"
nT<cr>	Type n lines
U<cr>	Change lowercase to uppercase (next entry)
V<cr>	Enable internal line number generation
nW<cr>	Write n lines to output file (start at beginning of buffer)
nX<cr>	Write next n lines to file "x$$$$$$$.LIB"
nZ<cr>	Pause n/2 seconds (2MHz)
n<cr>	Move forward n lines and type one line
<cr>	Move forward 1 line and type one line
-<cr>	Move backward and type one line
n:x<cr>	Move to n line number and perform "x" command
:mx<cr>	Perform command "x" from current line to line m
n: :mx<cr>	Move to n line number and perform command "x" through line number m

Note: "-" is valid on all positioning and display commands to indicate backward movement (e.g., -nD means to delete n characters behind the current position).

ERA

Erase file(s)
format: **ERA afn**<cr>
where: **afn** is an arbitrary file name with wild card matching.
action: The matching files are erased from the disk. If all files are specified (*.*), the question 'ALL FILES (Y/N)?' is displayed. Answer Y to erase all files, N to abort the command.
example: **A>ERA B:*.BAK**<cr>—erases all .BAK files from the "B" disk.

A>ERA PIP.COM<cr>— erases the file PIP.COM from the "A" disk.

MOVCPM

Configure a new CP/M

format: **MOVCPM (*¦ sz¦ sz *¦ * *)**<cr>

action: * indicates that CP/M is to calculate the correct size version of CP/M, configure it, move it to the correct memory location, and execute it

sz indicates that CP/M is to configure a szK byte version of CP/M, move it to the correct location, and execute it

sz * indicates that CP/M is to configure a szK byte version of CP/M, and then leave the new version in memory for a subsequent SAVE or SYSGEN operation

* * indicates that CP/M is to calculate and configure a version of CP/M that is correct for the present memory, and then leave it in memory for subsequent SAVE or SYSGEN operations

example: **A>MOVCPM 64 ***<cr>—configures a 64K version of CP/M. The new version is left in memory at location 100H. (See the SAVE command.)

PIP

Copy or transfer files

format: **PIP [pip-command]**<cr>

where: **pip-command** is dest=source[options]

dest is either a disk reference, an afn (no wild-cards), or any of the PIP logical devices.

source is either a disk reference, an afn (wild-cards allowed), or any of the PIP logical devices.

PIP logical devices are as follows:

CON — Console
LST — List device
PUN — Punch device
RDR — Reader device
INP — Patched character input
OUT — Patched character output
PRN — Same as LST device but with tabs

expanded to eight characters, lines numbered, and page breaks at 60 lines

EOF — Generates CP/M eof marker (Control Z)

NUL — Generates 40 nulls (used for punch device)

options are as follows (multiple options can be entered):

A — Archive, that is, only copy files that have been changed.

Dn — Delete (truncate) characters past column n.

E — Echo the file being transferred to the console.

F — Filter form feeds out of the file.

Gn — Get the source from user number n.

H — Ensure that valid Intel hex format is transferred.

I — Turn on "H" option and ignore :00 records

L — Translate upper- to lowercase.

N — Insert line numbers followed by a colon ahead of each line, starting at 1 and numbering by 1.

N2 — Insert line numbers followed by a colon ahead of each line, starting at 1, numbering by 1, include leading zeros, and place a tab character after each number.

O — Object file transfer. Ignore Control z's.

Pn — Set the page length to n, page eject every n lines.

Qs — Quit copying when the string s is found in the source file. When entering the string, terminate it with a control z.

Ss — Start the copy when string s is encountered in the source file. When entering the string s, terminate it with a control z.

Note: When strings are entered on the CP/M command line as PIP is invoked, automatic translation to uppercase is per-

formed. When strings are entered on the PIP command line, following the * prompt, no translation occurs.

Tn — Tabs are to be expanded to every nth character.

U — Lower- to uppercase translation is to occur.

V — Perform write verification

W — Write over read-only files

Z — Zero the parity bit of source characters.

action: **PIP<cr>** loads PIP, which will then issue the * prompt and wait for user commands. User commands are in the same form as commands entered on the CP/M command line.

PIP pip-command<cr> loads PIP, executes the pip-command, and then reboots the system.

example: (pip-commands)

A:=B:

Copy all files from the B disk to the A disk.

A:=B:FOO.*

Copy all files with filename FOO from the B disk to the A disk.

B:BIG=SMALL1.DOC,SMALL2.FOO

Concatenate files SMALL1.DOC and SMALL2.FOO (found on the current disk), copy them to the B disk, and name the resulting file BIG.

B:=A:PIP.COM[OV]

Copy the file PIP.COM from the A disk to the B disk. It's an object file and is to be write verified.

LST:=MYNOTES.DOC[T4P55]

List out the file MYNOTES.DOC from the currently logged disk. Expand tabs to every 4 characters and page break every 55 lines.

note: The destination disk *must* have been logged in *before* executing PIP. If not, you will receive the R/O error.

REN

Rename a file **(Built-in)**

format: **REN [disk reference]new = old<cr>**

where: **Disk reference** is optional. If omitted, the currently logged disk is assumed. Only one disk reference is allowed; that is, the old and new names must both refer to the same disk.

new is the new filename and suffix

old is the old filename and suffix (wildcards are not allowed)

action: The file specified in "old" is renamed "new." The "old" name is removed from the directory.

example:

A>REN B:NEWFILE.COM=OLDFILE.COM<cr>

—causes the file "OLDFILE.COM" on the B disk to be renamed "NEWFILE.COM". "OLDFILE.COM" no longer exists on B.

SAVE

Saves 256 byte pages on disk **(Built-in)**

format: **SAVE n afn<cr>**

where: **n** is the number of 256 byte (100H) pages to save

afn is the name given to the saved pages on disk

action: n pages of memory, starting at the Transient Program Area (TPA), usually location 100H, are saved on the named disk file. If a program was saved, it should be named as a .COM file for subsequent execution.

example: A>**SAVE 10 TEST.COM<cr>** saves 10 pages (2560 bytes) starting from the Transient Program Area (TPA) and labels it as TEST.COM.

STAT

Display or alter status information about files and peripheral devices **(Transient)**

DISPLAY STATUS

format: **STAT [d:¦ afn ¦ d:DSK:¦ DEV:¦ VAL:¦ USR:]<cr>**

where: No operands displays the status of the current disk

d: displays the status of a specific disk

afn displays the status and size of the named file(s) (wildcards are allowed)

d:DSK: displays drive characteristics of the named disk d:

DEV: displays the current logical to physical assignments of the devices CON:, RDR:, PUN:, and LST:

VAL: displays the possible logical to physical assignments and stat commands

USR: displays number of active files by user number

examples:

STAT VAL:<cr>—display the possible IO assignments and stat commands

STAT B:FOO.*<cr>—display the status of all files named FOO

STAT DEV:<cr>—display the current IOBYTE (logical to physical) device assignments

ALTER STATUS

format: **STAT [d:=R/O | afn $(R/O | R/W | SYS | DIR) ! logdev=phydev]**<cr>

where: **d:=R/O** changes the status of drive d: to Read/Only
(Control c resets it to Read/Write)

afn $(R/O | R/W | SYS | DIR) alters the status of the named file(s) (wildcards allowed) to:

R/O — Read Only
R/W — Read and Write
SYS — System file
DIR — Not a system file

logdev=phydev assigns the logical device logdev to the physical device phydev. This is the "IOBYTE" function.

examples:

A>**STAT B:=R/O**<cr>—sets drive B: to Read Only

A>**STAT FOO*.* $R/O**<cr>—sets all files on the current disk whose names begin with FOO to Read Only status

A>**STAT LST:=TTY:**<cr>—assigns the list device to the device on the TTY port

A>**STAT B:*.COM $SYS**<cr>—makes all .COM files on the B: disk into system files

SUBMIT

Submit a batch of CP/M commands for execution
(Transient)

format: **SUBMIT afn [parameters]**<cr>

where: **afn** is the name of a file with a suffix of .SUB
The file must contain CP/M commands, one per line.

parameters are positional parameters that are substituted for the $1, $2, etc., parameters used in the .SUB file.

action: SUBMIT builds a temporary file on the current disk and labels it $$$.SUB. The system is rebooted, and if the file $$$.SUB is present on the A disk, it is read as a source of input until all the commands in the file are exhausted. Processing of commands from the $$$.sub file can be aborted by typing the RUB character as commands are processed.

example: A file called EXAMPLE.SUB exists and has the following lines:

```
cc1 $1
clink $1 $2 $3
$1
```

The following SUBMIT command:

A>**SUBMIT EXAMPLE FOO ERRORS IOFILE**<cr>
generates the following $$$.SUB file:

```
CC1 FOO
CLINK FOO ERRORS IOFILE
FOO
```

which is stored on the current disk and the system is rebooted. If the current disk is the A disk, the commands in the $$$.SUB file are then executed.

SYSGEN

Copy the CP/M operating system from one disk to another **(Transient)**

format: **SYSGEN [afn]<cr>**

where: **afn** is the name of a file containing a CP/M operating system

action: If no afn is given, the following conversation takes place:

cpm: SYSGEN VERSION x.x
cpm: SOURCE DRIVE NAME (OR RETURN TO SKIP)
user: enter the drive that currently contains CP/M, such as **A**
cpm: SOURCE ON A, THEN TYPE RETURN
user: if you haven't already, put a disk that you know boots (contains a CP/M system) in the named drive (A, in this case) and type **<cr>**
cpm: FUNCTION COMPLETE (it has read the system)
cpm: DESTINATION DRIVE NAME (OR RETURN TO REBOOT)
user: enter the drive that will receive the CP/M system, such as **B**
cpm: DESTINATION ON B, THEN TYPE RETURN
user: if you haven't already, put the disk that you want to receive the CP/M system into the named drive (B, in this example), and press **<cr>**
cpm: FUNCTION COMPLETE (it has written the system on the destination disk)
cpm: DESTINATION DRIVE NAME (OR RETURN TO REBOOT) (this allows you to propagate the system to many disks without having to go through the SOURCE questions above every time)
user: continue as above until you've done all the disks you want, then type **<cr>**. The system will reboot.

TYPE

Type the contents of a file onto the console **(Built-in)**

format: **TYPE afn<cr>**

where: **afn** is the name of a file (no wildcards)

action: The contents of the file are displayed in ASCII on the console. Use **^S** to pause and restart the display. **^C** cancels the display. Use **^P** anywhere before the <cr> on the command line to echo the file to the LST: device.

example: **A>TYPE B:EXAMPLE.C<cr>** types the file B:EXAMPLE.C onto the console display.

A>TYPE BDSCIO.H P<cr> types the file "BDSCIO.H" onto the console and echoes it to the printer.

USER

Display or change the current user number **(Built-in)**

format: **USER [n]<cr>**

where: **n** is a user number from 0 to 15

action: If no user number is specified, the current user number is displayed. Otherwise, the current user number is set to n. (The default user number is 0.)

example: A>**USER 5<cr>** sets the user number to 5.
A>**USER<cr>** displays the current user number.

XSUB

Allowed buffered input in a submit file **(Built-in)**

format: **XSUB**

action: Placed as the first command in a submit file, XSUB allows subsequent commands in the submit file to read input from the submit file instead of from the console.

example: EXAMPLE.SUB contains the following commands:

```
XSUB
PIP
B:=A:*.C
B:=A:*.H
```

A>SUBMIT EXAMPLE<cr> causes the file "EXAMPLE.SUB" to be read as commands. XSUB causes the lines "B:=A:*.C" and "B:=A:*.H" to be read by PIP as if they had been typed at the console

CP/M Memory Organization (64K system)

NAME	ADDRESS	CONTENT DESCRIPTION
	FA00	The BIOS - Device dependent I/O routines
FBASE	EC00	The BDOS - Disk Operating System
CBASE	E400	The Console Command Processor (CCP)
TBASE	100	Transient Program Area (TPA)
DMA	080	Default disk I/O buffer (128 bytes)
	07D	Random record number (3 bytes)
FCB	05C	Default File Control Block (FCB) (32 bytes)
	006	Address of BDOS (2 bytes - HL)
	005	Jump to BDOS
	004	Left nybble = drive number Right nybble = current user number
	003	IOBYTE
	000	Jump to BIOS warm start

Error Messages—Fast Reference

During command and file processing, CP/M may write one of three errors to the console. They are:

BDOS ERR ON d: BAD SECTOR

CAUSE: This error message means that a physical error has occurred during the reading or writing of a diskette on the drive named in d:. It could be due to a faulty diskette or a failure in one of the disk read/write components. If the error persists from one disk to another, it's probably due to faulty electronics in your computer. If the error only shows up one diskette, it's probably a bad diskette.

RESPONSE: You can respond with a ^C to reboot, or a <cr>, which will cause the bad sector to be ignored and processing will continue. Be aware that if you respond with <cr>, data may be lost.

BDOS ERR ON d: SELECT

CAUSE: This error message means that a drive (d:) that is beyond the system capacity has been addressed. This is usually a program or user error.

RESPONSE: Striking any key on the console causes the system to warm boot. There is no other response.

BDOS ERR ON d: READ ONLY

CAUSE: This message indicates that an attempt has been made to write on a diskette which is in the "read/only" status. A drive is set to "read/only" status when the diskette in the drive is changed or is explicitly set to "read/only" by the STAT command. This message can be avoided by doing a control c (warm boot) after changing disks.

RESPONSE: Striking any key on the console causes the system to reboot. There is no other response. This means that if you are in the middle of a program, such as a text editor, everything you have done, but not saved, will be lost.

BDOS Calls—Fast Reference

The Basic Disk Operating System (BDOS) can be used by programmers for any of the 38 functions shown below (functions 28-40 are not valid for pre 2.0 versions. Functions after 36 are not valid for pre 2.2 versions). The functions are accessible by calling location 5, where there is a jump to the BDOS. The number of the desired function must be in the "C" register and any values passed to the function are passed in registers "DE" or sometimes just "E". Values returned by the function are in either register "A" or in registers "HL".

FUNC	FUNCTION NAME	INPUT PARAMETERS	OUTPUT RESULTS
0	System Reset	none	none
1	Console Input	none	A = char
2	Console Output	E = char	none
3	Reader Input	none	A = char
4	Punch Output	E = char	none
5	List Output	E = char	none
6	Direct Console I/O	E = FFH (input) E = Char (output)	A = Char or 00 if not ready
7	Get I/O Byte	none	A = IOBYTE
8	Set I/O Byte	E = IOBYTE	none
9	Print String	DE → Buffer	none
10	Read Console Buffer	DE → Buffer	Chars in Buffer
11	Get Console Status	none	A = 00/FF
12	Return Version Number	none	HL=Version #
13	Reset Disk System	none	none
14	Select Disk	E = Disk Number	none
15	Open File	DE → FCB	A = Dir Code
16	Close File	DE → FCB	A = Dir Code
17	Search for First	DE → FCB	A = Dir Code
18	Search for Next	none	A = Dir Code
19	Delete File	DE → FCB	A = Dir Code
20	Read Sequential	DE → FCB	A = Err Code
21	Write Sequential	DE → FCB	A = Err Code
22	Make File	DE → FCB	A = Dir Code
23	Rename File	DE → FCB	A = Dir Code
24	Return Login Vector	none	HL = Login Vect.
25	Return Current Disk	none	A = Cur Disk#
26	Set DMA Address	DE = DMA	none
27	Get Addr(Alloc)	none	HL=Alloc
28	Write Protect Disk	none	none
29	Get R/O Vector	none	HL= R/O Vector
30	Set File Attributes	DE → FCB	A = Dir Code
31	Get Addr(disk parms)	none	HL → DPB
32	Set/Get User Code	E = FFH/uc (get/set)	A = uc (get)
33	Read Random	DE → FCB	A = Err Code
34	Write Random	DE → FCB	A = Err Code
35	Compute File Size	DE → FCB (r0,rl,r2)	none
36	Set Random Record	DE → FCB (r0,rl,r2)	none
37	Reset Drive	DE → Drive vector	0
38 + 39 are unused			
40	Write random with zero fill	DE → FCB	A = Err code

Note that on return from BDOS calls, A = L, and B = H

Err codes for random mode:

- 01 — Trying to read unwritten data
- 02 — (Not returned)
- 03 — Cannot close current extent
- 04 — Seek to unwritten extent
- 05 — Directory overflow (Write mode only)
- 06 — Seek past physical end of disk

At last, it's easy to find all of the commands, options, syntaxes, and menus for running your most important software...all in one convenient Fast Reference Guide.

Alone, a Fast Reference Guide is a compact refresher course for the software. Bound into the original documentation, the Fast Reference Guide becomes an easy-to-find, easy-to-use glossary. Combined, the Fast Reference Guides for all your software make up a specially tailored system manual for *your* computer system.

Fast Reference Guides are prepared by MicroComputer Education, Inc.—specialists in teaching computing to users at every level, from beginners at home to veterans of mainframe programming.

The Fast Reference Guide series includes:

Fast Reference Guide to WordStar® (#1150)
Fast Reference Guide to VisiCalc® (#1151)
Fast Reference Guide to CP/M® (#1152)
Fast Reference Guide to SuperCalc® (#1153)
Fast Reference Guide to Using dBASE II® (#1154)
Fast Reference Guide to Programming dBASE II® (#1155)

HAYDEN BOOK COMPANY, INC.
Hasbrouck Heights, New Jersey

ISBN 0-8104-1152-0